

Design Assignment 2

Student Name: Ernesto Ibarra-Ayala

Student #: 2001211571

Student Email: ibarre3@unlv.nevada.edu

Primary Github address: <https://github.com/Ernesto-Ibarra/Work.git>

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

Atmel Studio 7.0

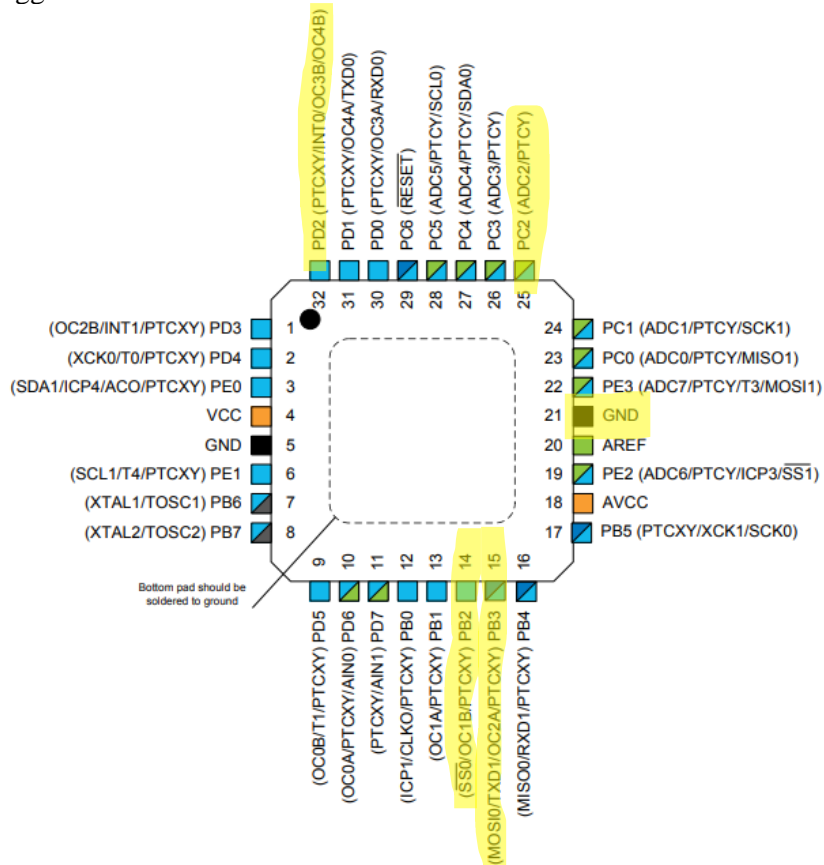
- Assembler
- Simulator
- Debugger

Atmega328PB-Xmini PC

Multi-Function Shield

- Switches
- LEDs

Logic Analyzer



2. DEVELOPED MODIFIED CODE OF TASK 1/2/3

Here is the code for all 3 tasks in Assembly.

```
.ORG 0
    jmp main
.ORG 0x02
    JMP EX0_ISR
main:
    LDI R20, HIGH(RAMEND) // initializing the stack
    OUT SPH, R20
    LDI R20, LOW(RAMEND)
    OUT SPL, R20

    SBI DDRB, 2 // makes portb.2 an output
    SBI DDRB, 3 // makes portb.3 an output
    SBI PORTB, 2 // set LED off
    SBI PORTB, 3 // set LED off

    CBI DDRC, 2 // makes portc.2 an input
    SBI PORTC, 2 // pulls up resistor pulled up
// Here is part 3 of the assignment
    LDI R20, 0x2
    STS EICRA, R20 // make it falling edge triggered
    SBI PORTD, 2 // enable pull up
    LDI R20, 1<<INT0
    OUT EIMSK, R20 // enable interrupt 0
    SEI // enable interrupts

// Here is Part 2 to turn on portb.2 for 1.25s after pressing a switch in portc.2
loop:
    ldi R22, 5
    SBIC PINC, 2 ; skip next instr if PINC is low
    jmp LED_ON ; jump when PINC is high
    CBI PORTB, 2 ; set LED on
    call delay_counter
    jmp loop

LED_ON:
    SBI PORTB, 2 ; set LED off
    jmp loop

// Here is Part 1 for a 0.25s Delay
delay_func:
    ldi R19, 21 ; R19 = 21
delay0:
    ldi R23, 255 ; R23 = 255
delay1:
    ldi R21, 248 ; R21 = 255
delay2:
    dec R21 ; Decrease R21 value
    brne delay2 ; if (R23 != 0) goto delay2 label
    dec R23 ; Decrease R23 value
    brne delay1 ; if (R23 != 0) goto delay1 label
    dec R19 ; Decrease R19 value
    brne delay0 ; if (R19 != 0) goto delay0 label
    nop
```

```

delay_counter: // this function will execute the 0.25s delay 5 times for the desired
1.25s
    dec R22
    brne delay_func
    ret

EXO_ISR:
    CBI PORTB, 3 // turn on port3
    ldi R22,2 // here we load 2 into r22 so it makes a 0.5sec delay
    call delay_counter // here we call the delay function
    SBI PORTB, 3 // here we turn off led 3
    RETI // we return.

```

Here is the code for task 1/2/3 in C.

```

#define F_CPU 16000000UL
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>

int main ()
{
    DDRB |= (1<<3); // PB3 as an output

    PORTB |= (1<<3); // turn off the LED

    DDRC &= (0<<2); //make portC.2 an input
    PORTC |= (1 <<2); //enable pull up

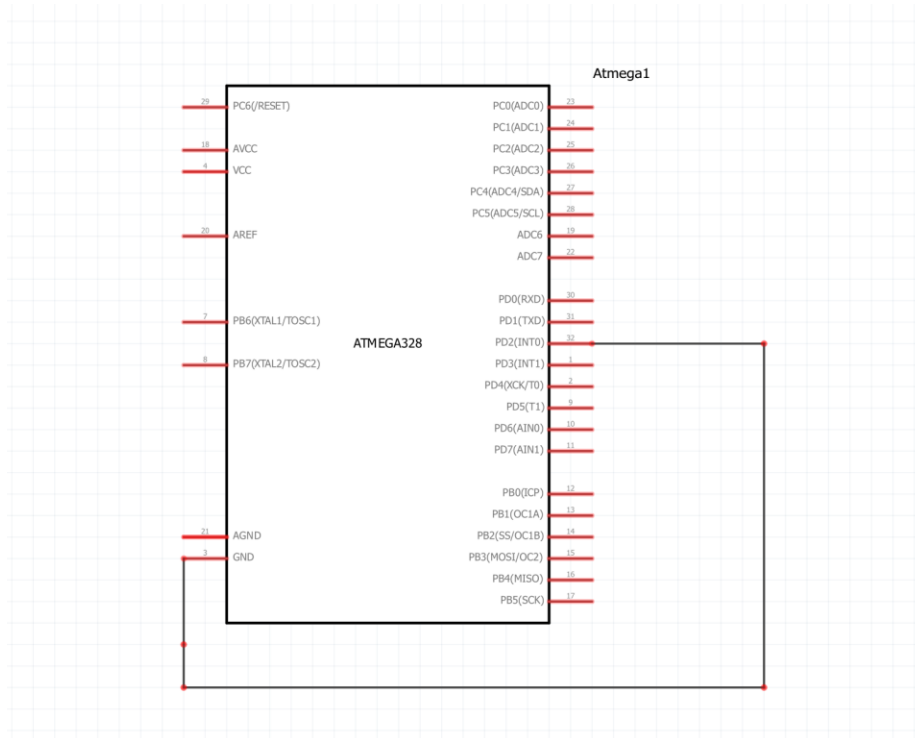
    PORTD |= (1<<2); //pull-up activated
    EICRA = 0x02; //make INT0 falling edge triggered
    EIMSK = (1<<INT0); //enable external interrupt 0
    sei (); //enable interrupts

    while (1)
    {
        if(!(PINC & (1<<PINC2))) // Here we are doing Part2 of the assignment
        {
            PORTB &= ~(1<<3); // turns on
            _delay_ms(1250);
        }
        else
        {
            PORTB |= (1<<3); // turns off
        }
    }
}

ISR (INT0_vect) //ISR for external interrupt 0, here we are doing part 3 of the
assignment.
{
    PORTB &= (0<<3); //turn on PORTB.3
    _delay_ms(500);
}

```

3. SCHEMATICS



4. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

Here is the output for assembly for task 1/2/3. I will show everything working in the videos down below. I had to use PORTC2 instead of 3 because I think my PORTC3 is not working properly. So I just used the second switch.

DA2 (Debugging) - Microchip Studio

Advanced Mode Quick Launch (Ctrl+Q)

File Edit View VAssistX ASF Project Build Debug Tools Window Help

Debug

Debug Browser

delay_func_5

ATmega328PB

debugWIRE on mEDBG (ATML2523052700003994)

Registers

R00 = 0x64 R01 = 0x00 R02 = 0xFF R03 = 0xFF R04 = 0xFF R05 = 0xFF R06 = 0xFF R07 = 0xFF R08 = 0xFF R09 = 0xFF R10 = 0xFF R11 = 0xFF

R12 = 0xFF R13 = 0xFF R14 = 0xFF R15 = 0xFF R16 = 0xFF R17 = 0xFF R18 = 0xFF R19 = 0x00 R20 = 0x01 R21 = 0x00 R22 = 0x05 R23 = 0x00

R24 = 0x00 R25 = 0x00 R26 = 0x03 R27 = 0x01 R28 = 0x11 R29 = 0x00 R30 = 0x60 R31 = 0x00

main.asm

```

STS EICRA, R20 //mkae it falling edge triggered
SBI PORTD, 2 //enable pull up
LDI R20, 1<INT0
OUT EIMSK, R20 // enable interrupt 0
SEI // enable interrupts

// Here is Part 2 to turn on portb.2 for 1.25s after pressing a switch in portc.2
loop:
ldi R22,5
SBI PINC, 2 ; skip next instr if PINC is low
jmp LED_ON ; jump when PINC is high
CBI PORTB, 2 ; set LED on
call delay_counter
jmp loop

LED_ON:
SBI PORTB, 2 ; set LED off
jmp loop

// Here is Part 1 for a 0.25s Delay
delay_func:
ldi R19,21 ; R19 = 21
delay0:
ldi R23,255 ; R23 = 255
delay1:
ldi R21,248 ; R21 = 255
delay2:
dec R21 ; Decrease R21 value
brne delay2 ; if (R23 != 0) goto delay2 label
dec R23 ; Decrease R23 value
brne delay1 ; if (R23 != 0) goto delay1 label
dec R19 ; Decrease R19 value
brne delay0 ; if (R19 != 0) goto delay0 label
nop

```

I/O

Filter:

AC

Analog-to-Digital Convert...

CFD

CPU

EEPROM

EXINT

PORTB

PORTC

PORTD

PORTE

SPIO

SPI1

TC0

Name	Address	Value	Bits
PINB	0x23	0xBF	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
DDRB	0x24	0x0C	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
PORTB	0x25	0x0C	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>

Watch 1

Name	Value	Type

Memory 1

Memory: prog FLASH

Address: 0x0000,prog

prog 0x0000	0c 94 04 00 0c 94 2e 00 48 e0 4e bf 4f ef 4d bf 22 9a	..
prog 0x0012	23 9a 2a 9a 2b 9a 3a 98 42 9a 42 e0 40 93 69 00 5a	9a #3*
prog 0x0024	41 e0 4d bb 78 94 65 e0 32 99 0c 94 1e 00 2a 98 0e	94 AaM
prog 0x0036	2b 00 0c 94 15 00 2a 9a 0c 94 15 00 35 e1 7f ef 58 ef	+. .
prog 0x0048	5a 95 f1 f7 7a 95 d9 f7 3a 95 c1 f7 00 00 6a 95 a1 f7	Z.ñ
prog 0x005A	08 95 2b 98 62 e0 0e 94 2b 00 2b 9a 18 95 2f 00 2b	9a ..+
prog 0x006C	18 95 64 00 0c 94 64 00 0c 94 64 00 0c 94 64 00 0c	94 ..d
prog 0x007E	64 00 0c 94 64 00 0c 94 64 00 0c 94 64 00 0c 94 64	00 d..
prog 0x0090	0c 94 64 00 0c 94 64 00 0c 94 64 00 0c 94 64 00 0c	94 ..d

Autos

Locals

Watch 1

Watch 2

Memory 1

Call Stack

Breakpoints

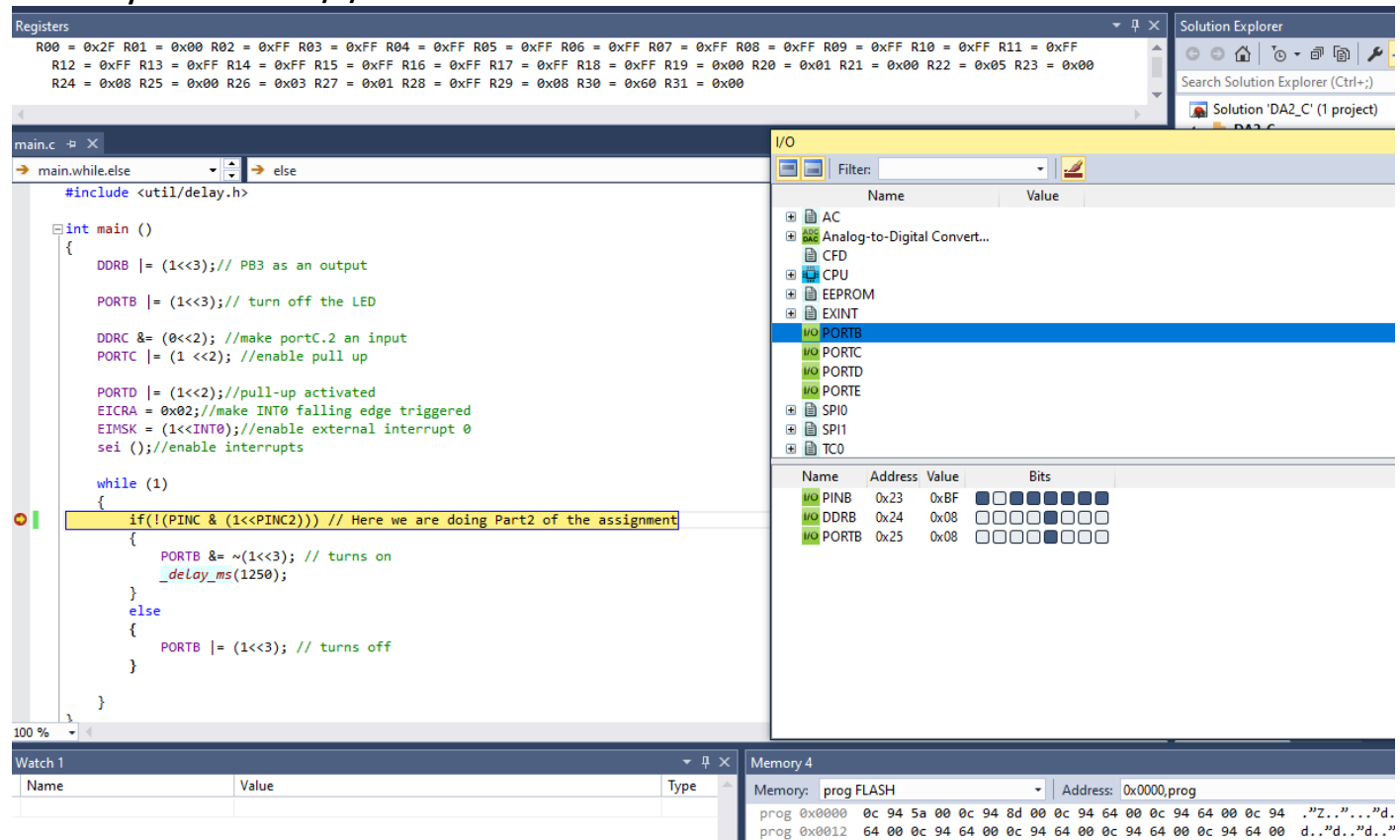
Command Window

Immediate Window

Output

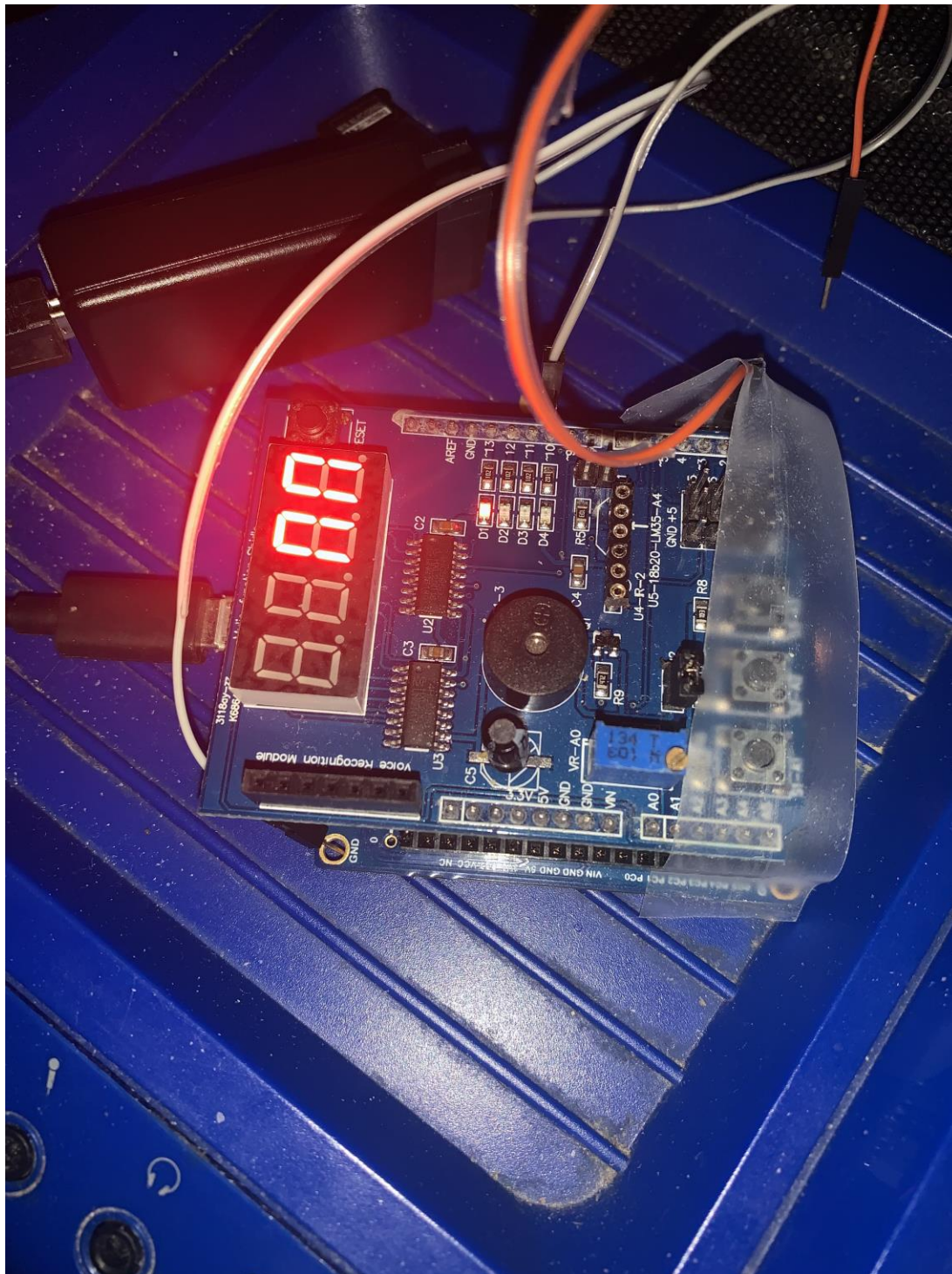
Memory

Here is my code for task 1/2/3 in C.



5. SCREENSHOT OF EACH DEMO (BOARD SETUP)

Here is a screenshot for both assembly and C since they both do the same thing. Here is a picture of the LED turning on when I press the button. I will show a video displaying how the interrupt works and the delays. Here you can see the cable being connected to PD2 and another cable being connected to PORTB2. That cable will be connected to the logic analyzer so we can analyze the delay, another cable is connected to ground. I have some tape since there is a couple loose pins, I need to make sure are tied down.



6. VIDEO LINKS OF EACH DEMO

This is video for Task 1/2/3

<https://www.youtube.com/watch?v=4kFULTV545I>

This is video for Logic Analyzer

<https://www.youtube.com/watch?v=G-iQ5USn8MY>

7. GITHUB LINK OF THIS DA

<https://github.com/Ernesto-Ibarra/Work/tree/main/DesignAssignments>

Student Academic Misconduct Policy

<http://studentconduct.unlv.edu/misconduct/policy.html>

"This assignment submission is my own, original work".

Ernesto Ibarra